

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-267960

(43)Date of publication of application : 29.09.2000

(51)Int.Cl.

G06F 13/00
H04L 12/56

(21)Application number : 11-076377

(71)Applicant : LION CORP
CSK CORP

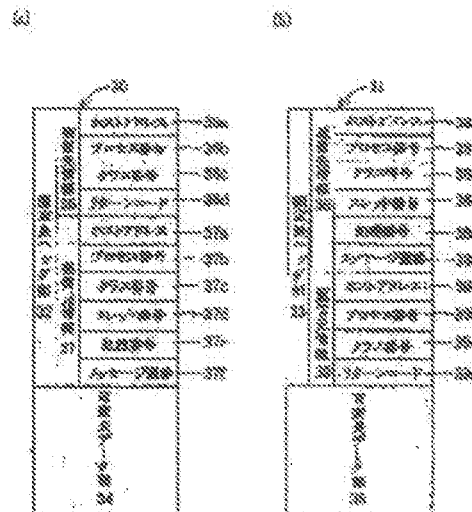
(22)Date of filing : 19.03.1999

(72)Inventor : SATO TOYOYUKI
KOBAYASHI AKITO(54) METHOD FOR COMMUNICATING PACKET BETWEEN PLURAL PROCESSES AND
PACKET COMMUNICATION EQUIPMENT

(57)Abstract:

PROBLEM TO BE SOLVED: To easily transmit/receive a packet between plural processes and to easily prepare a program in an inter-process packet communication equipment using a prescribed packet for the transmission/reception of a message between processes.

SOLUTION: A prescribed transmission packet 30 is transmitted by a class instance of a transmission source process and a transmitted process receiving the packet 30 accesses an instance specified by the packet 30, and when a replay is requested by the packet 30, sends a prescribed reply packet 31. A replied process receiving the packet 31 accesses the class instance of the transmission source process specified by the packet 31.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

(18) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-267960

(P2000-267960A)

(43) 公開日 平成12年9月29日 (2000.9.29)

(51) Int.Cl. ⁷	識別記号	F I	チ-コ-ド [*] (参考)
G 0 6 F 13/00	3 5 3	G 0 6 F 13/00	3 5 3 A 5 B 0 8 9
H 0 4 L 12/58		H 0 4 L 11/20	1 0 2 A 5 K 0 3 0
			9 A 0 0 1

審査請求 未請求 請求項の数 5 O L (全 10 頁)

(21) 出願番号 特願平11-76377

(22) 出願日 平成11年3月19日 (1999.3.19)

(71) 出願人 000000793

ライオン株式会社

東京都墨田区本所1丁目3番7号

(71) 出願人 000131291

株式会社シーエスケイ

東京都新宿区西新宿2丁目6番1号

(72) 発明者 佐藤 豊之

東京都墨田区本所1丁目3番7号 ライオン株式会社内

(74) 代理人 100078835

弁護士 村田 幹雄

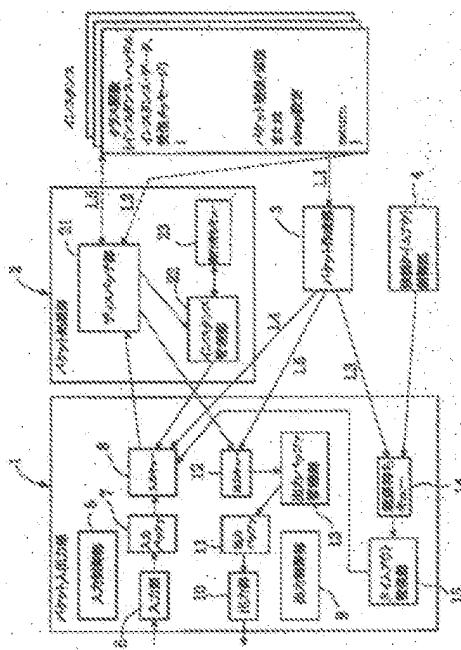
最終頁に続く

(54) 【発明の名称】 プロセス間のパケット送信方法及びパケット送信装置

(57) 【要約】

【課題】 プロセス間のメッセージの送受信に所定のパケットを使用するプロセス間のパケット通信装置において、プロセス間で容易にパケットの送受信を実現し、かつプログラムの作成を容易にすることを目的とする。

【解決手段】 発信元プロセスのクラスインスタンスにおいて、所定の発信パケット30を発信し、該発信パケット30を受信した発信先プロセスにおいて、発信パケット30により特定されるインスタンスを呼び出し、発信パケット30により返信が要求されている場合に、所定の返信パケット31を返信し、該返信パケット31を受信した返信先プロセスにおいて、返信パケット31により特定される発信元プロセスのクラスインスタンスを呼び出すことを特徴とする。



【特許請求の範囲】

【請求項1】 プロセス間のメッセージの送受信に所定のバケットを使用するプロセス間のバケット通信方法であって、

発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するか否かを示す情報と、を含む発信バケットを発信し、

前記発信バケットを受信した発信先プロセスにおいて、前記発信バケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信バケットにより返信が要求されている場合に、前記発信元プロセスのクラスインスタンスを特定する情報を含む返信バケットを返信し、前記返信バケットを受信した返信先プロセスにおいて、前記返信バケットにより特定される発信元プロセスのクラスインスタンスを呼び出す、
ことを特徴とするプロセス間のバケット通信方法。

【請求項2】 プロセス間のメッセージの送受信に所定のバケットを使用するプロセス間のバケット通信方法であって、

発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するか否かを示す情報と、を含む発信バケットを発信し、該発信バケットを返信待ちバケットとして保持すると共に発信先プロセスからの返信を待たずに他の処理を行い、

前記発信バケットを受信した発信先プロセスにおいて、前記発信バケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信バケットにより返信が要求されている場合に、前記発信元プロセスのクラスインスタンスを特定する情報を含む返信バケットを返信し、前記返信バケットを受信した返信先プロセスにおいて、該返信バケットの識別データと一致する識別データを有する前記返信待ちバケットにより特定される発信元プロセスのクラスインスタンスを呼び出す、
ことを特徴とするプロセス間のバケット通信方法。

【請求項3】 プロセス間のメッセージの送受信に所定のバケットを使用するプロセス間のバケット通信方法であって、

発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するか否かを示す情報と、返信を受信した後に前記発信元プロセスにおいて行う処理を特定する情報と、を含む発信バケットを発信し、該発信バケットを返信待ちバケットとして保持すると共に発信先プロセスからの返信を待たずに他の処理を行い、

前記発信バケットを受信した発信先プロセスにおいて、

前記発信バケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信バケットにより返信が要求されている場合に、前記発信元プロセスのクラスインスタンスを特定する情報と、返信を受信した後に前記発信元プロセスにおいて行う処理を特定する情報と、を含む返信バケットを返信し、前記返信バケットを受信した返信先プロセスにおいて、該返信バケットの識別データと一致する識別データを有する前記返信待ちバケットにより特定される発信元プロセスのクラスインスタンスの処理を行う、
ことを特徴とするプロセス間のバケット通信方法。

【請求項4】 複数の接続相手とのメッセージの送受信に所定のバケットを使用するバケット通信装置であって、

バケット通信の接続相手ごとに、該接続相手から入力されるバケットを格納する入力バッファと、該接続相手から出力するバケットを格納する出力バッファと、を有するチャンネルを備えてなり共に、接続相手から接続要求があった場合に、前記チャンネルのうちの空きチャンネルを該接続相手に割り当てる接続管理手段と、

入力可能状態である前記チャンネルの入力バッファにバケットを入力する入力制御手段と、

出力可能状態である前記チャンネルの出力バッファからバケットを出力する出力制御手段と、
を備えてなることを特徴とするバケット通信装置。

【請求項5】 前記チャンネルにおいて入力されたバケットを一時的に退避させる入力キューと、前記チャンネルから出力するバケットを一時的に退避させる出力キューと、

前記出力キュー内のバケットに対応するチャンネルが空き状態の場合に、該出力キューに入っているバケットを該チャンネルの出力バッファに出力する出力バッファ管理手段と、

を備えてなることを特徴とする請求項4に記載のバケット通信装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、プロセス間のメッセージの送受信に所定のバケットを使用するプロセス間のバケット通信装置において、プロセス間で容易にバケットの送受信を実現し、かつプログラムの作成を容易にすることができるプロセス間のバケット通信方法及びバケット通信装置に関するものである。

【0002】

【従来の技術】同一コンピュータ内又は異なるコンピュータ内に分散配置されたオブジェクト間の通信においては、RPC(Remote Procedure Call)をベースとした技術を用いてシステム開発が行われている。各種ラングーション・モニタ、オブジェクト・リクエスト・プロ

一方等にもRPC又はRPCと同様の技術が使われ、メッセージによる処理の呼び出しが行われている。また、上記メッセージを複数のコンピュータに配信あるいは複数のコンピュータから収集する場合には、スイッチングハブ等に代表される集配装置や、各コンピュータからのメッセージを順に入力し、それぞれメッセージの送信先に配信するシステムが用いられる。

【0002】

【発明が解決しようとする課題】しかしながら、上述のRPCを用いたシステム開発においては、共通言語で記述されたインタフェース部分のプログラムをアプリケーション側の言語に適合させるため、ブリコンパイルが必要であり、必ずしも使いやすい開発環境とはいえなかった。また、送信したメッセージに対する返信を待つ場合の処理の方法として同期型と非同期型とがあるが、同期型では、返信メッセージを受信するまで処理が止まってしまう他の処理を行うことができない。一方、非同期型では、返信メッセージを受信するまで処理が止まってしまうことはないが、返信されたメッセージを処理するためのプログラムが必要になり、また返信されたメッセージを処理すると共に他の処理を行うためには、複雑なプログラミングが要求される。ここで、非同期型において返信されたメッセージを処理すると共に他の処理を行うプログラミングを簡単に実現する方法として、マルチプロセス又はマルチスレッドによる方法がある。しかし、マルチプロセスによる方法ではコンピュータ資源を合計に使ってしまうという問題があり、マルチスレッドによる方法ではメモリの排他管理や通信のための複雑管理等が必要になり、プログラムが複雑になってしまうという問題がある。

【0004】また、上述のようにメッセージを複数のコンピュータに配信等する場合にスイッチングハブ等を用いると、2つのコンピュータ間で通信が行われている間は、その他のコンピュータに対する通信が行われず、その他のコンピュータにおける処理は待ち状態になる。更に、各コンピュータからのメッセージを順に入力し、それぞれメッセージの送信先に配信するシステムでは、1のコンピュータからのメッセージを全て入力してから次のコンピュータからのメッセージを入力し、1のコンピュータへのメッセージを全て出力してから次のコンピュータへのメッセージを出力するという方法のため、大きなメッセージを入出力する場合、あるいは通信速度が遅いコンピュータと通信する場合には、他のコンピュータにおける通信が待たされ、システム全体の性能が低下することになる。

【0005】本発明は、このような従来の問題点を鑑みてなされたもので、プロセス間のメッセージの送受信に所定のバケットを使用するプロセス間のバケット通信装置において、プロセス間で容易にバケットの送受信を実現し、かつプログラムの作成を容易にすることができる

プロセス間のバケット通信方法及びバケット通信装置を提供することを目的とする。

【0006】

【課題を解決する手段】このような従来のプロセス間のバケット通信方法における問題点を解決するために、請求項1に記載の本発明は、プロセス間のメッセージの送受信に所定のバケットを使用するプロセス間のバケット通信方法であって、発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するかどうかを示す情報と、を含む発信バケットを発信し、前記発信バケットを受信した発信先プロセスにおいて、前記発信バケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信バケットにより返信が要求されている場合に、前記発信元プロセスのクラスインスタンスを特定する情報を含む返信バケットを返信し、前記返信バケットを受信した返信先プロセスにおいて、前記返信バケットにより特定される発信元プロセスのクラスインスタンスを呼び出すことを特徴としている。

【0007】また、請求項2に記載の本発明は、発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するかどうかを示す情報と、を含む発信バケットを発信し、該発信バケットを返信待ちバケットとして保持すると共に発信先プロセスからの返信を待たずに他の処理を行い、前記発信バケットを受信した発信先プロセスにおいて、前記発信バケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信バケットにより返信が要求されている場合に、前記発信元プロセスのクラスインスタンスを特定する情報を含む返信バケットを返信し、前記返信バケットを受信した返信先プロセスにおいて、該返信バケットの識別データと一致する識別データを有する前記返信待ちバケットにより特定される発信元プロセスのクラスインスタンスを呼び出すことを特徴とする。

【0008】さらに請求項3に記載の本発明は、発信元プロセスのクラスインスタンスにおいて、発信先のプロセス及びクラスを特定する識別データと、該発信元プロセスのクラスインスタンスを特定する情報と、返信を要求するかどうかを示す情報と、返信を受信した後に前記発信元プロセスにおいて行う処理を特定する情報と、を含む発信バケットを発信し、該発信バケットを返信待ちバケットとして保持すると共に発信先プロセスからの返信を待たずに他の処理を行い、前記発信バケットを受信した発信先プロセスにおいて、前記発信バケットにより特定される発信先クラスのインスタンスを呼び出し、前記発信バケットにより返信が要求されている場合に、前記発信元プロセスのクラスインスタンスを特定する情報

と、返信を受信した後に前記返信元プロセスにおいて行う処理を特定する情報と、を含む返信パケットを返信し、前記返信パケットを受信した返信先プロセスにおいて、該返信パケットの識別データと一致する識別データを有する前記返信待ちパケットにより特定される返信元プロセスのクラスインスタンスの処理を行うことを特徴とする。

【0009】また、請求項4に記載の本発明は、複数の接続相手とのメッセージの送受信に所定のパケットを使用するパケット通信装置であって、パケット通信の接続相手ごとに、該接続相手から入力されるパケットを格納する入力バッファと、該接続相手に出力するパケットを格納する出力バッファと、を有するチャンネルを備えてなると共に、該接続相手から接続要求があった場合に、前記チャンネルのうちの空きチャンネルを該接続相手に割り当てる接続管理手段と、入力可能状態である前記チャンネルの入力バッファにパケットを入力する入力制御手段と、出力可能状態である前記チャンネルの出力バッファからパケットを出力する出力制御手段と、を備えてなるとことを特徴として構成されている。

【0010】さらに請求項5に記載の本発明は、前記請求項4に記載の本発明において、前記チャンネルにおいて入力されたパケットを一時的に遅延させる入力キューと、前記チャンネルから出力するパケットを一時的に遅延させる出力キューと、前記出力キュー内のパケットに対応するチャンネルが空き状態の場合に、該出力キューに入っているパケットを該チャンネルの出力バッファに入力する出力バッファ管理手段と、を備えてなるとことを特徴として構成されている。

【0011】

【発明の実施の形態】以下、本発明の第一の実施形態について図面を参照して詳細に説明する。図1は、本実施形態におけるプロセス間パケット通信装置の一部の構成を示す図、図2は、本実施形態におけるパケットのデータ構成を示す図、図3はパケット通信の手順を示す図である。本実施形態におけるプロセス間パケット通信装置は、同一又は異なるコンピュータのプロセス内のオブジェクト間で所定のパケットを用いてメッセージを交換するものである。ここで各オブジェクトは、プロセス内のクラスとして定義される。プロセス内には1つ又は複数のクラスが定義され、1つのクラスが1つのサービスを提供する。また各クラスは各種の処理を実行する関数等で実装される。各クラスには1つのクラス番号が付与され、コンピュータのホストアドレス、プロセス番号、及びクラス番号を特定することによりクラスが一意に識別される。また各クラスにおいてはインスタンスが生成され、このインスタンスに対してクラス毎にスレッド番号が付与される。従って、クラス番号とスレッド番号とからなるインスタンス・ハンドルによってインスタンスが一意に識別される。更に各クラスには、インスタンスが

存在する間にインスタンス固有のデータを保存するためのメモリ領域(インスタンス・データ領域、図示せず)を定義することができる。このインスタンス・データ領域については後述する。

【0012】図1に示すように本実施形態におけるプロセス間パケット通信装置は、パケット入出力部1、パケット処理部2、パケット作成部3、強制タイムアウト制御部4とを各プロセスごとに備えてなる。パケット入出力部1は、パケットの入出力処理とパケット返信のタイムアウトを制御するものであり、入力制御部5、入力部6、入力バッファ7、入力キュー8、出力制御部9、出力部10、出力バッファ11、出力キュー12、出力バッファ管理部13、返信待ちキュー14、及びタイムアウト監視部15を備えてなる。パケット入出力部1の入力制御部5は、入力可能状態となっている入力部6を呼び出し、該入力部6にデータを入力する。またパケット入出力部1の入力部6は、一定時間内に入力したデータ又は一定量のデータを入力バッファ7に入力する。そして、1つのパケットの入力が完了すると、入力バッファ7内のデータを入力キュー8に出力し、入力部6は次のパケットの入力可能状態となる。パケット入出力部1の出力制御部9は、出力バッファ11にデータがあるときに、出力可能状態となっている出力部10を呼び出して出力バッファ11内のデータを出力する。またパケット入出力部1の出力バッファ管理部13は、出力バッファ11にデータがないときに、出力キュー12に格納されたパケットを取り出して出力バッファ11に入れる。パケット入出力部1の出力部10は、一定時間内に出力するデータ又は一定量のデータを出力バッファ11から取り出し、出力する。そして、1つのパケットの出力が完了すると、出力バッファ11は空き状態となり、出力部10は出力待ち状態となる。またパケット入出力部1のタイムアウト監視部15は、返信待ちキュー14に格納された返信待ち情報のタイムアウト時間を監視する。

【0013】またパケット処理部2は、パケット入出力部1において送受信されるパケットに基づいてクラスインスタンスの生成と消滅を制御するもので、ディスパッチ部21、インスタンス管理部22、及び実行待ちキュー23を備えてなる。パケット処理部2のディスパッチ部21は、パケット入出力部1の入力キュー8からパケットを取り出し、該パケットによって特定される返信先のプロセス内のクラスのインスタンスを起動又は消滅させる。また、インスタンス管理部22は、後述する最大スレッド数に基づいてインスタンスの多重度を管理する。パケット作成部3は、プロセス間の通信に用いられるパケットの固定長ヘッダ部と可変長部を作成する。パケットの送信先が返信元と異なる他のプロセスであるときは、作成したパケットを出力キュー12に入れる。一方、パケットの送信先が、送信元と同じプロセスであるときは、作成したパケットを直接入力キュー8に入れ、

送信する無駄を省いている。また強制タイムアウト制御部4は、返信待ち情報を強制的にタイムアウトさせるものである。

【0014】次に、本実施形態にて使用するパケットについて説明する。図2(a)は本実施形態にて使用する発信パケットのデータ構成を示す図。(b)は返信パケットのデータ構成を示す図である。本実施形態にて使用するパケットとしては、メッセージを発信する発信パケット30と該メッセージに対する返信メッセージを返信する返信パケット31の2種類がある。これら発信パケット、返信パケットともに、図2に示すように発信先(あるいは返信先)と発信元(あるいは返信元)を特定する固定長ヘッダ部32、33とメッセージを格納した可変長データ部34、35とから構成される。図2(a)に示すように、発信パケット30の固定長ヘッダ部32は、発信先情報36、発信元情報37から構成される。この発信先情報36及び発信元情報37にはそれぞれコンピュータのホストアドレス36a、プロセス番号36b、クラス番号36cからなる識別データが含まれ、これらの識別データにより発信先あるいは発信元のプロセス及びクラスが特定される。また発信先情報36は、上記識別データに加えてリターンコード36dを含んでおり、このリターンコード36dには、パケット送信中にエラーが起きたときにエラーコードが設定される。発信パケット30における発信元情報37は、上記識別データ37a〜37cに加えて発信元のクラスインスタンスのスレッド番号37d、処理番号37e、及びメッセージ通番37fを含んでいる。このスレッド番号37dと上記発信元の識別データ37a〜37cとにより、発信元のクラスインスタンスを一意に識別することができる。また図2(b)に示すように、返信パケット31における返信先情報38は、識別データ38a〜38c、スレッド番号38d、処理番号38e、及びメッセージ通番38fを含んでおり、返信元情報39には、識別データ39a〜39c及びリターンコード39dを含んでいる。即ち、返信パケット31の固定長ヘッダ部33は、発信パケット30の発信先情報36と発信元情報37を、返信元情報39と返信先情報38とに入れ替えたものである。

【0015】次に、本実施形態に係るプロセス間パケット通信装置におけるパケット通信の手順について説明する。まず、発信パケットの発信の手順について説明する。あるクラスから他のクラス(発信元クラスと同一のプロセス内のクラスであってもよいし、異なるプロセス内のクラスであってもよい。また、他のクラスとあるが、発信元クラスと同一のクラスであってもよい。)にメッセージを送信する際には、パケット作成部3を呼び出し(L1)、発信する発信パケットを作成する。この発信パケットの固定長ヘッダ部32の発信先情報36には、発信先クラスのホストアドレス、プロセス番号、及

びクラス番号が設定され、発信元情報37には、発信元クラス自身のホストアドレス、プロセス番号、クラス番号、及び該クラスインスタンスのスレッド番号等が設定される。この発信パケットの固定長ヘッダ部32により、発信先のクラス及び発信元のクラスインスタンスを一意に識別することができる。

【0016】ここで発信元のクラスインスタンスが発信先クラスからの返信を要求しない場合には、発信パケットの固定長ヘッダ部32の発信元情報37の処理番号37eに0を設定する。一方、発信先クラスからの返信を要求する場合には、上記処理番号37eに0以外の値を設定する。即ち、この発信元情報37の処理番号37eは返信を要求するか否かを示す情報として機能する。

【0017】発信パケットの発信元情報37の処理番号37eに0以外の値を設定し、発信先クラスからの返信を要求する場合には、発信パケットを発信したクラス側から0以外の返信値にて戻る(L2)。このとき発信元のクラスインスタンスはそのまま存続し、返信待ち状態となる。この返信待ち状態となる際に、発信した発信パケットのコピー(返信待ちパケット)及び返信待ち状態のタイムアウト時間が、返信待ち情報としてパケット入出力部1の返信待ちキュー14に保存される(L3)。なお、この返信待ち状態のタイムアウト時間は、各パケットごとに異なる値を指定することができる。

【0018】ここで、パケット入出力部1のタイムアウト監視部15は、この返信待ち情報のタイムアウト時間を監視する。タイムアウトした場合は、タイムアウト・エラー・コードをリターンコード39dに設定した返信パケットが、パケット作成部3により作成される。その後、このタイムアウト・エラーの返信パケットを入力キュー8に入れる(L4)と共に、該返信待ち情報が削除される。なお、このタイムアウト・エラーの返信パケットを(出力キュー12ではなく)直接入力キュー8に入れることで、返信パケットを送信する無駄を省いている。また、上記パケット入出力部1の返信待ちキュー14に保存された返信待ち情報は、強制タイムアウト制御部4によってクラスインスタンス単位あるいはクラス単位で又は全てを強制的にタイムアウトさせることができる。このときには、タイムアウト・エラー・コードの替わりに任意のリターンコード39dを設定することができる。なお、発信パケットの発信先をNULクラスに設定すると、実際には発信パケットは発信されず、上記返信待ち情報のみがパケット入出力部1の返信待ちキュー14に保存される。そして、該返信待ち情報のタイムアウト時間後に対応するクラスインスタンスが呼び出されることとなり、クラスインスタンスを一定時間休止(sleep)させ、一定時間間隔での繰り返し処理が可能となる。

【0019】一方、発信パケットの発信元情報37の処理番号37eに0を設定し、発信先クラスからの返信を

要求しない場合には、発信バケットを発信したクラス関数から返値0にて戻り(L2)、バケット処理部2のディスパッチ部2.1がこの発信元のクラスインスタンスを消滅させる。バケット作成部3により作成された発信バケットは、バケット入出力部1の出力キュー1.2に入れられ、出力バッファ1.1、出力部1.0を介して発信先のプロセスに送られる。上述のように、返信を要求する場合は、クラスインスタンスの中からバケットを発信する必要があるが、返信を要求しないのであれば、発信元のクラスインスタンスを特定するクラス番号とスレッド番号は特別の意味を持たないため、クラス番号とスレッド番号に0以外の値を設定すれば、クラスインスタンスの外からバケットを発信することも可能である。

【0020】次に、上記発信バケットの受信及び返信バケットの返信の手順について説明する。発信バケットが発信先プロセスで受信されると、発信バケットの固定長ヘッダ部3.2の発信先情報3.6で特定した発信先クラスのインスタンスがバケット処理部2のディスパッチ部2.1により生成される。そして該クラスインスタンスが呼び出され、発信バケットのメッセージが渡される(L5)。なお受信メッセージを受け取ったクラスインスタンスは、1つの監視スレッドとして起動され、インスタンスを示すハンドルを引数にしてクラス関数が呼ばれる。

【0021】ここで受信した発信バケットの発信元情報3.7の処理番号3.7aが0以外の値の場合、即ち発信元のクラスインスタンスが返信を要求する場合には、バケット作成部3を呼び出し(L1)、返信する返信バケットを作成する。この返信バケットの固定長ヘッダ部3.3の返信元情報3.9には受信した発信バケットの発信元情報3.6が設定され、返信先情報3.8には受信した発信バケットの発信元情報3.7が設定される。これによりメッセージを発信したクラスインスタンスに返信バケットが返送されることとなる。発信バケットと同様に、作成された返信バケットは、バケット入出力部1の出力キュー1.2に入れられ(L6)、出力バッファ1.1、出力部1.0を介して返信先のプロセスに送られる。

【0022】一方、受信した発信バケットの発信元情報3.7の処理番号3.7aが0の場合、即ち発信元のクラスインスタンスが返信を要求しない場合には、上記返信バケットは返信されない。なお発信バケットを受信した発信先プロセスにおいて、発信バケットの固定長ヘッダ部3.2の発信先情報3.6で特定したクラスが登録されていない場合は、その旨のエラーコードをリターンコード3.9dに設定した返信バケットがバケット作成部3により作成され、該返信バケットが発信元クラスに返送される。

【0023】次に返信バケットの受信の手順について説明する。返信バケットが発信元プロセスで受信されると、受信された返信バケットの返信先情報3.8のクラス

番号3.9c、スレッド番号3.8d、メッセージ番号3.8fと上述のバケット入出力部1の返信待ちキュー1.4に保存された返信待ちバケットの発信元情報3.7のクラス番号3.7c、スレッド番号3.7d、メッセージ番号3.7f番を比較し、これらすべてが一致する返信待ちバケット(即ち、返信バケットに対応する返信待ちバケット)を検索する。対応する返信待ちバケットが見つければ、該返信待ちバケットの固定長ヘッダ部3.2の発信元情報3.7で特定されるクラス、即ち発信バケットを発信したクラスインスタンスが再度呼び出され、該返信待ちバケットに対応する返信待ち情報が削除される。このとき返信バケットの返信先情報3.8の処理番号3.8e(換言すれば、発信バケットの発信元情報3.7の処理番号3.7e)に対応する処理が行われる。従って発信バケットの発信元情報3.7の処理番号3.7eは、上述のように返信を要求するかどうかを示す情報として機能すると共に、返信したときに行う処理を特定する情報としても機能する。

【0024】更に上記処理番号は、クラスインスタンス内で、発信バケットにて呼び出されたのか、返信バケットにて呼び出されたのかを識別するためにも使用することができる。つまり、クラスインスタンスに渡す処理番号を返信用処理番号と処理識別用処理番号の2つに分け、返信用処理番号には、発信バケットの発信元情報3.7の処理番号3.7eを入れ、処理識別用処理番号には、発信バケットのときには0を、返信バケットのときには返信先情報3.9の処理番号3.8eを入れる。返信先情報3.8の処理番号3.8eは0になることがないため、処理識別用処理番号が0のときは、発信バケットで呼び出されたことになり、0以外のときは返信用バケットで呼び出されたことになる。なお、このクラスインスタンスの呼び出しに際し、上述のインスタンス・データ領域に保存したデータを使用して前回呼び出し時の処理を継続することができる。

【0025】この場合において、あるクラスインスタンスが返信待ち状態であっても、他のクラスインスタンスに対する返信バケットが受信されると対応するそのクラスのインスタンスが起動されるが、このクラスインスタンスが終了するか又は返信待ち状態になれば、該返信待ち状態のクラスインスタンスに対する返信バケットの受信により該返信待ちのクラスインスタンスが再度起動されることになる。一方、対応する返信待ちバケットが見つからなければ、警告メッセージをログに出力し受信した返信バケットは破棄される。なお、受信したバケットが発信バケットあるいは返信バケットのいずれであるかは、バケットの固定長ヘッダ部のスレッド番号の設定によって判別される。即ち、バケットを受信したプロセスは、受信したバケットの発信先情報3.6(あるいは返信先情報)にスレッド番号が設定されていないならば発信バケットを受信したと判断し、発信先情報3.6(あるいは返信先情報)にスレッド番号が設定されているならば

11

返信パケットを受信したと判断する。

【0026】ここでクラスごとの最大スレッド数又は全クラス合計の最大スレッド数を指定することができる。パケット処理部2のインスタンス管理部22は、パケットを受信してもクラスインスタンスを作成せず、該受信したパケットをインスタンス管理部22内の実行待ちキュー23に入れる。この実行待ちキュー23に入れられたパケット（実行待ちパケット）は、クラスインスタンスが削減したときに取り出され、該パケットに対応するインスタンスが生成される。即ち、クラスインスタンスが削減したときには、そのクラスのスレッド数と全クラスのスレッドの合計が1つ減るので、実行待ちパケットの中で、対応するクラスが最大スレッド数に達しておらず、かつ最先に入った実行待ちパケットが取り出される。このように最大スレッド数を指定可能とすることによって、クラスごとの多重度又は全クラスの多重度を制御でき、コンピュータ資源の消費及び性能を容易にコントロールすることができる。

【0027】次に、上述したパケット通信の手順を具体例を挙げて説明する。図3は本実施形態に係るプロセス間パケット通信装置におけるパケット通信の手順を示したものである。以下、図3を参照して説明する。まずプロセス101からプロセス102に発信パケットを発信する（S1）。このときの発信パケットの発信元情報37には「プロセス番号=101、処理番号=0」が設定され、発信先情報36には「プロセス番号=102、クラス番号=1」が設定されている。このプロセス101のプロセスから発信されたパケットはプロセス管理部100を介してプロセス102に送られる（S2）。なおプロセス管理部100は、プロセスの起動・終了、接続受付・登録・抹消の管理や、各プロセスの監視、各プロセスへの通知、各プロセス間におけるパケットの中継等の機能を有するものである。

【0028】プロセス102では、パケット入出力部1によって上記発信パケットが受信され、パケット処理部2によって、発信パケットの発信先情報36によって特定されるクラス番号1のクラスインスタンスが生成される（S3）。このとき処理識別用処理番号は0となるので、クラス1内の処理番号0の処理が実行される。なお、返信用処理番号は0であるので発信元のプロセス101には返信しない。上記プロセス102のクラス1内の処理では、更に発信パケットが発信される（S4）。この発信パケットの発信先情報36には「プロセス番号=103、クラス番号=3」が設定され、発信元情報37には「プロセス番号=102、クラス番号=1、処理番号=10」が設定される。ここで、この発信パケットは「処理番号=10」であり、返信を要求するものである。従って返信1にて該クラス関数から戻り（S5）、返信待ち状態になる。

【0029】一方、上記プロセス102から発信された

12

発信パケットはプロセス管理部100を介してプロセス103のパケット入出力部1により受信される。そしてプロセス103のパケット処理部2によりクラス番号3のクラスインスタンスが生成される（S6）。このクラス内の関数では返信パケットが受信される（S7）。この返信パケットの返信先情報には、受信した発信パケットの発信元情報37、即ち「プロセス番号=102、クラス番号=1、処理番号=10」が設定され、返信元情報には、受信した発信パケットの発信先情報36、即ち「プロセス番号=103、クラス番号=3」が設定される。このプロセス103から返信された返信パケットは、プロセス管理部を介してプロセス102のパケット入出力部1により受信される。そして、プロセス102のパケット処理部2によりクラス番号1のクラスインスタンスが再度呼び出され（S8）、該クラス1内の処理番号10の処理が実行される。

【0030】プロセス102のクラス1内の処理番号10の処理においては、プロセス104へ発信パケットを発信する（S9）。このときの発信パケットの発信先情報36は「処理番号=0」とされ、返信を待たないので該クラス関数から返信0にて戻る（S10）。このように各プロセス間におけるメッセージの送受信が可能とされる。

【0031】次に本発明の第二の実施形態について図面を参照して詳細に説明する。なお特に説明しない部分については上記第一の実施形態と同様である。図4は本実施形態におけるプロセス間パケット通信装置の構成の一部を示す図である。図4に示すように、本実施形態におけるプロセス間パケット通信装置のパケット入出力には、接続するコンピュータごとに1つの接続チャネルが設けられている。それぞれのチャネルには、入力部8、入力バッファ7、及び出力バッファ11が設けられる。なお、この入力バッファ7及び出力バッファ11は、1つのパケットの全てを格納できる大きさを持つ。

【0032】ここで、本実施形態におけるプロセス間パケット通信装置には、本装置への接続要求（又は切断要求）を監視する接続管理部50が設けられている。この接続管理部50は、接続要求があれば接続し、接続先を空きチャネルに割り当てる。逆に、切断要求があれば接続を切断し、チャネルの割り当てを解除する。そして、接続される相手方のコンピュータからは、一定時間内に入力できたデータ又は一定量のデータ（パケット）がそれぞれ入力され、入力バッファ7に入れられる。また、接続される相手方のコンピュータには、一定時間内に出力できるデータ又は一定量のデータ（パケット）が出力バッファ11から取り出され出力される。1つのパケットの入力が完了すると、入力バッファ7内を入力キュー8に出力し、入力部8は次のパケットの入力待ち状態（入力可能状態）となる。また、1つのパケットの出力が完了すると、出力バッファ11は空状態になる。

【0033】バケット処理部2は、入力キュー8からバケットを取り出し、バケットの固定長ヘッダ部にある発信先情報（返信先情報）に対応する出力チャネル情報を付加して出力キュー12に入れる。出力バッファ管理部13は、出力バッファ11に入っているバケットに対応するチャネルの出力バッファ11が空状態のときは、そのバケットを出力キュー12から取り出して該空状態の出力バッファ11に入れる。出力制御部9は、出力バッファ11にバケットが残っている（出力可能状態）チャネルの出力部10を順次呼び出してデータを出力する。一方、入力制御部5は、入力可能状態となっているチャネルの入力部6を順次呼び出してバケットを入力する。なお、各チャネルには優先順位を付けることができ、特定のチャネルの通信を優先させることが可能である。入力部6と出力部10とを異なる優先順位とすることもでき、また同一の優先順位とすることもできる。

【0034】

【発明の効果】上述したように本発明は、プロセス間のメッセージの送受信に所定のバケットを使用するプロセス間のバケット通信方法であって、発信元プロセスのクラスインスタンスにおいて、所定の発信バケットを発信し、該発信バケットを受信した発信先プロセスにおいて、発信バケットにより特定されるクラスインスタンスを呼び出し、返信が要求されている場合に、所定の返信バケットを返信し、該返信バケットを受信した返信先プロセスにおいて、前記返信バケットにより特定されるクラスインスタンスを呼び出すことにより、RPCを用いなくともプロセス間にてメッセージの送受信が可能となるので、プログラムを作成する際にプリコンパイルをする必要がなくなり、容易にプログラムを作成することができるといふ効果がある。

【0035】また発信元プロセスのクラスインスタンスにおいて、所定の発信バケットを発信し、該発信バケットを返信待ちバケットとして保持すると共に発信先プロセスからの返信を待たずに他の処理を行い、発信バケットを受信した発信先プロセスにおいて、発信バケットにより特定される発信先クラスのインスタンスを呼び出し、発信バケットにより返信が要求されている場合に、所定の返信バケットを返信し、該返信バケットを受信した返信先プロセスにおいて、該返信バケットの識別データと返信待ちバケットとに基づいて発信元プロセスのクラスインスタンスを呼び出すことにより、発信バケットに対する返信を待つ場合であっても、返信バケットを受信したときに処理が呼び出されるので、返信を待っている間において他の処理を行うことができ、かつ、このための特別なプログラムを必要としないといふ効果がある。また、返信バケットには、返信後の処理を特定する情報を含めることとしたので、返信後の処理を容易に特定してプログラムを作成することができるといふ効果がある。さらにマルチプロセスではなく、1つのプロセス

内の機能としてバケットの送受信を行うことができるので、コンピュータの資源を余計に使うことがないといふ効果がある。また、OSレベルで制御を行うマルチスレッドではなく、関数レベルでマルチスレッドを実現している（擬似マルチスレッド）ので、メモリの排他的管理や通信のための接続管理等のプログラムを必要としないといふ効果がある。

【0036】さらに本発明は、複数の接続相手とのメッセージの送受信に所定のバケットを使用するバケット通信装置であって、バケット通信の接続相手ごとに入力バッファと出力バッファとを有するチャネルと、接続相手から接続要求があった場合に、前記チャネルのうちの空きチャネルを該接続相手に割り当てる接続管理手段と、入力可能状態である前記チャネルの入力バッファにバケットを入力する入力制御手段と、出力可能状態である前記チャネルの出力バッファからバケットを出力する出力制御手段と、を備えてなることにより、1つのチャネルに対して1つバケットの入出力が完了する前に、他のチャネルの処理を行うことができ、バケットの大きさ又は通信速度の違いに影響されずに各チャネルに対する平均的な処理速度を保証できるという効果がある。

【0037】またチャネルにおいて入力されたバケットを一時的に退避させる入力キューと、チャネルから出力するバケットを一時的に退避させる出力キューと、出力キュー内のバケットに対応するチャネルが空状態の場合に、該出力キューに入っているバケットを該チャネルの出力バッファに入力する出力バッファ管理手段と、を備えてなることにより、各チャネルの入力と出力及び配信処理が相互の処理状態の影響を受けずに動作するので、全体としてのスループットが向上するといふ効果がある。

【図面の簡単な説明】

【図1】本発明の第一の実施形態におけるプロセス間バケット通信装置の一部の構成を示す図である。

【図2】本発明の第一の実施形態におけるバケットのデータ構成を示す図で、(a)は発信バケットのデータ構成を示す図、(b)は返信バケットのデータ構成を示す図である。

【図3】本発明の第一の実施形態におけるバケット通信の手順を示す図である。

【図4】本発明の第二の実施形態におけるプロセス間バケット通信装置の構成の一部を示す図である。

【符号の説明】

- 1 バケット入出力部
- 5 入力制御部
- 7 入力バッファ
- 8 入力キュー
- 9 出力制御部
- 11 出力バッファ

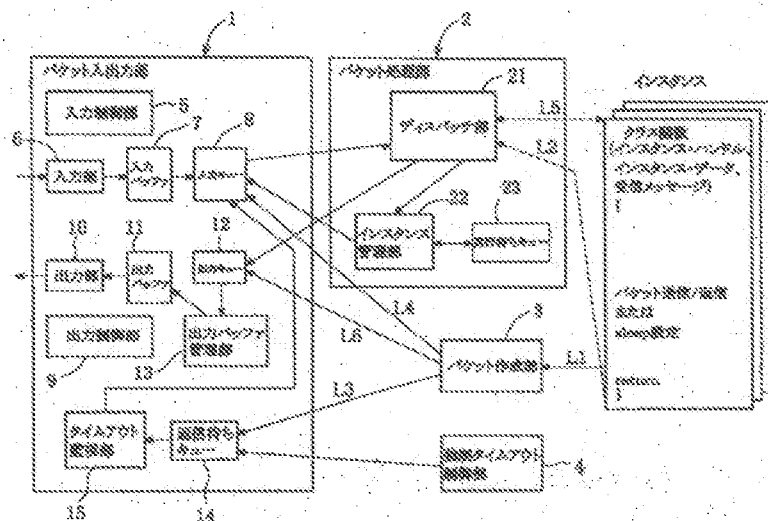
22

22

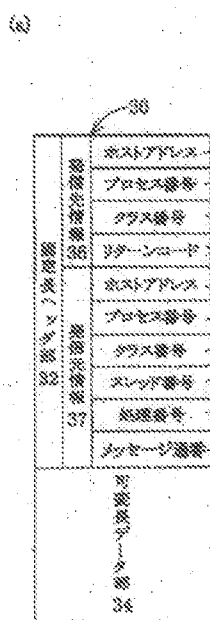
- | | |
|----|--------------|
| 12 | 出力キュー |
| 13 | 出力バッファ管理部 |
| 2 | パケット 処理部 |
| 21 | ディスパッチ部 |
| 3 | パケット 作成部 |
| 4 | 強制タイムアウト 制御部 |

- | | |
|----|---------|
| 30 | 発信バケット |
| 36 | 発信先情報 |
| 37 | 発信元アドレス |
| 31 | 返信バケット |
| 38 | 返信先情報 |
| 39 | 返信元情報 |

THE



182

**INDEX**